UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/547,288	04/11/2000	Nir N. Shavit	1004-4663	4871

22120 7590 11/19/2004

ZAGORIN O'BRIEN & GRAHAM, L.L.P.
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/547,288	SHAVIT ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 August 2004 and 26 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-43 have been considered.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 2, 4-9, 25-31, and 36-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacGregor et al., U.S. Patent Number 4,584,640 (herein referred to as MacGregor) in view of Greenspan et al., U.S. Patent Number 6,128,710 (herein referred to as Greenspan) and in view of Arnold, U.S. Patent Number 5,081,572 (herein referred to as Arnold).

4. Referring to claim 1, MacGregor has taught a method of managing access to an array susceptible to concurrent operations on sequence encoded therein, the method comprising executing as part of a pop operation, an dual target compare and swap (DCAS) to update a then-current, end identifying index for the array and an element of the array adjacent to that identified by the end identifying index (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B). MacGregor has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing

data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

5. In addition, MacGregor has not taught returning from the DCAS, on failure thereof, an indication by which an empty state of the array is detectable. Arnold has taught returning from the DCAS, on failure thereof, an indication by which an empty state of the array is detectable (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

6. Referring to claim 2, MacGregor has not taught wherein the indication by which the empty state of the array is detectable is indicative of presence of a distinguishing value in the adjacent element. Arnold has taught wherein the indication by which the empty state of the array is detectable is indicative of presence of a distinguishing value in the adjacent element (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the

invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

7. Referring to claim 4, MacGregor has taught

   a. Wherein the pop operation is a left pop operation (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

   b. Wherein the end identifying index is a left-end index (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

   c. Wherein the adjacent element is to the right of the identified element (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

8. Referring to claim 5,

   a. Wherein the pop operation is a right pop operation (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

   b. Wherein the end identifying index is a right-end index; and wherein the adjacent element is to the left of the identified element (MacGregor Abstract; column 1,

line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

9. Referring to claim 6, MacGregor has taught a method of managing access to an array susceptible to concurrent operations on a sequence encoded therein, the method comprising:

a. Executing as part of a push operation, an dual target compare and swap (DCAS) to update a then-current, end identifying index for the array and an element of the array identified by the end identifying index (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

10. MacGregor has not taught an atomic compare and swap. Greenspan has taught an atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

11. In addition, MacGregor has not taught returning from the DCAS, on failure thereof, an indication by which a full state of the array is detectable. Arnold has taught returning from the DCAS, on failure thereof, an indication by which a full state of the array is detectable (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13;

Figure 4A; Figure 4B; Figure 6; and Figure 8). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

12. Referring to claim 7, MacGregor has not taught wherein the indication by which the full state of the array is detectable is indicative of absence of a distinguishing value in the identified element. Arnold has taught wherein the indication by which the full state of the array is detectable is indicative of absence of a distinguishing value in the identified element (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

13. Referring to claim 8, MacGregor has taught

a. Wherein the push operation is a left push operation (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. Wherein the end-identifying index is a left-end index (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

14. Referring to claim 9, MacGregor has taught

a. Wherein the pop operation is a right push operation (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. Wherein the end-identifying index is a right-end index (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

15. Referring to claim 25, MacGregor has taught a method of managing concurrent access to an array susceptible to competing accesses at same and opposing ends thereof, the method comprising:

a. Executing as part of a first access operation, an dual target compare and swap (DCAS) to update a first end identifying index and an element of the array corresponding to a then-current value thereof (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

b. Executing as part of a competing second access operation, a DCAS to update a second end identifying index and an element of the array corresponding to a then-current value thereof (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B),

16. MacGregor has not taught an atomic compare and swap. Greenspan has taught an atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

17. In addition, MacGregor has not taught wherein, if successful completion of one of the first and the second competing access operations results in a boundary condition state of the array, the DCAS of the other of the first and the second access operations fails and returns an indication thereof. Arnold has taught wherein, if successful completion of one of the first and the second competing access operations results in a boundary condition state of the array, the DCAS of the other of the first and the second access operations fails and returns an indication thereof (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the

empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

18. Referring to claims 26 and 29, MacGregor has taught

a. Wherein the first access operation and the competing second access operation are competing pop operations (Applicant's claim 26) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

b. Wherein the array elements corresponding to the first and second indices are each adjacent to that identified by the respective index (Applicant's claim 26) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

c. Wherein the first access operation and the competing second access operation are competing push operations (Applicant's claim 29) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

d. Wherein the array elements corresponding to the first and second indices are each identified by the respective index (Applicant's claim 29) (MacGregor Abstract;

column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

19. MacGregor has not taught

a. Wherein the boundary condition state is an empty state (Applicant's claim 26);

b. Wherein the adjacent element referenced by the failing one of the competing pop operations encodes a distinguishing value signifying the empty state (Applicant's claim 26);

c. Wherein the boundary condition state is an full state (Applicant's claim 29); and

d. Wherein the array element referenced by the failing one of the competing push operations encodes a value other than a distinguishing value (Applicant's claim 29).

20. Arnold has taught

a. Wherein the boundary condition state is an empty state (Applicant's claim 26) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8);

b. Wherein the adjacent element referenced by the failing one of the competing pop operations encodes a distinguishing value signifying the empty state (Applicant's claim 26) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8);

c. Wherein the boundary condition state is an full state (Applicant's claim 29) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8); and

d. Wherein the array element referenced by the failing one of the competing push operations encodes a value other than a distinguishing value (Applicant's claim 29) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8).

21. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

22. Referring to claims 27, 28, 30, and 31, MacGregor has taught

a. Wherein the competing pop operations are competing same end pop operations (Applicant's claim 27) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. Wherein the first index and the second index identify a same end of the array (Applicant's claim 27) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line

39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

c. Wherein the competing pop operations are competing same end pop operations (Applicant's claim 28) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

d. Wherein the first index and the second index identify a same end of the array (Applicant's claim 28) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

e. Wherein the competing push operations are competing opposing end push operations (Applicant's claim 30) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

f. Wherein the first index and the second index identify opposing ends of the array (Applicant's claim 30) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

g. Wherein the competing push operations are competing same end push operations (Applicant's claim 31) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

h. Wherein the first index and the second index identify a same end of the array (Applicant's claim 31) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

23. Referring to claim 36, MacGregor has taught a concurrent shared object implementation comprising:

a. A contiguous array encoded in an addressable store (MacGregor column 2, line 61 to column 3, line 5 and Figure1);

b. Opposing indices into the contiguous array usable to delimit there between a portion of the contiguous array for storage of a sequence of zero or more data values (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

c. A computer readable encoding of push and pop operations defined to operate or elements of the contiguous array and on respective of the opposing indices

(MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B),

d. Wherein the push operation employs a first instance of a dual target compare and swap (DCAS) operation to update one of the opposing indices (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B) and

e. Wherein the pop operation employs a second instance of a DCAS operation to update one of the opposing indices and a corresponding element of the contiguous array (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B)

24. MacGregor has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

25. In addition, MacGregor has not taught

a. A corresponding element of the contiguous array while returning on failure, an indication by which a full state of the contiguous array is detected, and

b. While returning on failure, an indicator by which an empty state of the contiguous array is detected.

26. Arnold has taught

a. A corresponding element of the contiguous array while returning on failure, an indication by which a full state of the contiguous array is detected (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8), and

b. While returning on failure, an indicator by which an empty state of the contiguous array is detected (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8).

27. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

28. Referring to claim 37-39, MacGregor has taught

a. Wherein concurrent shared object includes a deque (Applicant's claim 37) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-

42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. Wherein the computer readable encoding of push and pop operations includes:

i. Opposing end variants of the pop operation (Applicant's claim 37) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

ii. Opposing end variants of the push operation (Applicant's claim 37) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

c. Wherein concurrent shared object includes a queue or FIFO (Applicant's claim 38) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

d. Wherein the computer readable encoding of push and pop operations operate on opposing ends of the queue or FIFO (Applicant's claim 38) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

e. Wherein concurrent shared object includes a stack or LIFO (Applicant's claim 39) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

f. Wherein the computer readable encoding of push and pop operations operate on a same end of the stack or LIFO (Applicant's claim 39) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

29. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over MacGregor in view of Greenspan in view of Arnold as applied to claim 1 above, and further in view of Mark Allen Weiss's Data Structures and Algorithm Analysis in C++ Second Edition © 1999 (herein referred to as Weiss). MacGregor has not taught wherein the array encodes a double-ended queue as a circular buffer of bounded size, the end-identifying index and an opposing end identifying index delimiting the sequence. Weiss has taught wherein the array encodes a double-ended queue as a circular buffer of bounded size, the end-identifying index and an opposing end identifying index delimiting the sequence (Weiss page 111). A person of ordinary skill in the art would have recognized, and as taught by Weiss, a circular queue lets elements be added to the beginning of a queue after they have been enqueued, thereby allowing more elements to enter the queue even after all positions have been used at one point. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Weiss in MacGregor.

30. Claims 10-16 and 20-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnold, U.S. Patent Number 5,081,572 (herein referred to as Arnold) in view of Greenspan et al., U.S. Patent Number 6,128,710 (herein referred to as Greenspan).

31. Referring claim 10, Arnold has taught a method of providing concurrent access to a double ended data structure of bounded size implemented using a circular buffer technique, the method comprising:

a. As part of an access to a first-end of the double-ended data structure, performing in alternate legs of a conditional branch (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B):

b. A first multi-way compare and swap on then-current contents of a first-end index store and a corresponding element of the double ended data structure to disambiguate a retry state and a boundary condition state of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B);

c. A second multi-way compare and swap on then-current contents of the first-end index store and a corresponding element of the double-ended data structure the second multi-way compare and swap performing the access and, on failure thereof, returning an indication disambiguating a retry state and the boundary condition state of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B),

d. Wherein the conditional branch discriminates between presence and absence of a distinguishing value in an element of the double-ended data structure

corresponding to the then-current contents of the first-end index store (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

32. Arnold has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of Arnold to maintain data coherency.

33. Referring to claim 11, Arnold has taught

a. Wherein the access includes a pop from the first-end of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B);

b. Wherein the boundary condition state is an empty state of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B); and

c. Wherein the retry state results from a concurrently performed push or pop access at the first-end of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

34. Referring to claim 12, Arnold has taught

a. Wherein the access includes a push onto the first-end of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B);

b. Wherein the boundary condition state is a full state of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B); and

c. Wherein the retry state results from a concurrently performed push or pop access at the first-end of the double-ended data structure (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

35. Referring to claim 13, Arnold has taught wherein the double-ended data structure includes a double-ended queue (deque) (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

36. Referring to claim 14, Arnold has taught wherein the multi-way compare and swap is a dual target_compare and swap (DCAS) (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

37. Referring to claim 15, Arnold has taught a method of managing concurrent access to double-ended queue (deque), the method comprising:

a. Employing, in an implementation of a pop operation, execution of a dual target compare and swap (DCAS) to interrogate instantaneous values of a first end index and a deque element adjacent to that identified thereby for a signature indicative of an empty state of the array, the signature including presence in that adjacent

element of a distinguishing value (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B),

b. Wherein successful execution of an opposing end pop operation includes execution of a DCAS to update a second end index and a deque element adjacent to that identified thereby, the update of that adjacent element storing the distinguishing value therein (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

38. Arnold has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of Arnold to maintain data coherency.

39. Referring to claim 16, Arnold has taught wherein successful execution of a competing, same end pop operation includes execution of a DCAS to update the first end index and a deque element adjacent to that identified thereby, the update of that adjacent element storing the distinguishing value therein (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

40. Referring to claim 20, Arnold has taught

a. Employing, in an implementation of a push operation, execution of an dual target compare and swap (DCAS) to interrogate instantaneous values of a third end

index and a deque element identified thereby for a signature indicative of an full state of the deque, the signature including absence in that identified deque element of a distinguishing value (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B),

b. Wherein successful execution of an opposing end push operation includes execution of a DCAS to update a fourth end index and a deque element identified thereby, the update of the identified deque element storing a value other than the distinguishing value therein (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

41. Arnold has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of Arnold to maintain data coherency.

42. Referring to claim 21, Arnold has taught

a. Wherein the first end index and the third end index identify a same end of the deque (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B); and

b. Wherein the second end index and the fourth end index identify a same end of the deque (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

43. Referring to claim 22, Arnold has taught

a. Wherein the first end index and the fourth end index identify a same end of the deque (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B); and

b. Wherein the second end index and the third end index identify a same end of the deque (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

44. Referring to claim 23, Arnold has taught a method of managing concurrent access to double ended queue (deque), the method comprising:

a. Employing, in an implementation of a push operation, execution of an dual target compare and swap (DCAS) to interrogate instantaneous values of a first end index and a deque element identified thereby for a signature indicative of a full state of the deque, the signature including absence in that identified deque element of a distinguishing value (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B),

b. Wherein successful execution of an opposing end push operation includes execution of a DCAS to update an opposing end index and a deque element identified thereby, the update of the identified deque element storing a value other

than the distinguishing value therein (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

45. Arnold has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of Arnold to maintain data coherency.

46. Referring to claim 24, Arnold has taught wherein successful execution of a competing, same end push operation includes execution of a DCAS to update the first end index and a deque element identified thereby, the update of that adjacent element storing a value other than the distinguishing value therein (Arnold column 9, lines 23 to column 10, line 59; Figure 4A; and Figure 4B).

47. Claims 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnold in view of Greenspan as applied to claim 15 above, and further in view of MacGregor et al., U.S. Patent Number 4,584,640 (herein referred to as MacGregor). Arnold has not taught

a. Wherein the first end index is a left index and, if the state of the deque is non-empty, the deque element adjacent to that identified thereby is a left most element of the deque (Applicant's claim 17);

b. Wherein the second end index is a right index and, if the state of the deque is nonempty, the deque element adjacent to that identified thereby is the right most element of the deque (Applicant's claim 17).

c. Wherein the pop operation is a left pop operation and the opposing end pop operation is a right pop operation (Applicant's claim 18); and

d. Wherein the first end index is a left end index and the element adjacent to that identified thereby is adjacent to the right (Applicant's claim 18).

e. Wherein the distinguishing value is encoded as a null value (Applicant's claim 19).

48. MacGregor has taught

a. Wherein the first end index is a left index and, if the state of the deque is non-empty, the deque element adjacent to that identified thereby is a left most element of the deque (Applicant's claim 17) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

b. Wherein the second end index is a right index and, if the state of the deque is nonempty, the deque element adjacent to that identified thereby is the right most element of the deque (Applicant's claim 17) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

c. Wherein the pop operation is a left pop operation and the opposing end pop operation is a right pop operation (Applicant's claim 18) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

d. Wherein the first end index is a left end index and the element adjacent to that identified thereby is adjacent to the right (Applicant's claim 18) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

e. Wherein the distinguishing value is encoded as a null value (Applicant's claim 19) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

49. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by MacGregor, these operations improve linked list manipulation. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the operations of MacGregor in Arnold to improve linked list manipulation.

50. Claims 32, 35, 40, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacGregor et al., U.S. Patent Number 4,584,640 (herein referred to as MacGregor) in view of Greenspan et al., U.S. Patent Number 6,128,710 (herein referred to as Greenspan) and in view of

Mark Allen Weiss's Data Structures and Algorithm Analysis in C++ Second Edition © 1999 (herein referred to as Weiss).

51. Referring to claim 32, Arnold has taught a double-ended queue (deque) implementation comprising:

a. A contiguous array S of bounded size encoded in an addressable store (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

b. A left index L and a right index R into the contiguous array, the contiguous array S, the left index L and the right index R together defining a buffer with state including a sequence of zero or more values encoded in the contiguous array between elements S[L] and S[R] thereof and wherein at least the S[L] and S[R] entries encode a distinguishing value (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B);

c. A computer readable encoding of at least a first access operation, execution of the first access operation operating at a particular end of the sequence and employing a dual target compare and swap (DCAS) to update a corresponding one, but not both, of the left and right indices Land R and an element of the contiguous array adjacent to the contiguous array element identified thereby (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53

to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

52. MacGregor has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

53. In addition, MacGregor has not taught wherein the array encodes a double-ended queue as a circular buffer of bounded size, the end identifying index and an opposing end identifying index delimiting the sequence. Weiss has taught wherein the array encodes a double-ended queue as a circular buffer of bounded size, the end-identifying index and an opposing end identifying index delimiting the sequence (Weiss page 111). A person of ordinary skill in the art would have recognized, and as taught by Weiss, a circular queue lets elements be added to the beginning of a queue after they have been enqueued, thereby allowing more elements to enter the queue even after all positions have been used at one point. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Weiss in MacGregor.

54. Referring to claim 35, MacGregor has taught

a. Computer readable encodings of at least three additional access operations (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-

42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B),

b. Wherein the first and three additional access operations include push and pop operations at left and rights end of the sequence, respectively (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

55. Referring to claim 40, MacGregor has taught a computer program product encoded in at least one computer readable medium, the computer program product comprising:

a. Instances of the at least one functional sequence concurrently executable by plural processors of a multiprocessor and each including a dual target compare and swap (DCAS) to update a corresponding one of the end identifying indices and an element of the array corresponding to a then-current value thereof (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. The DCAS of the at least one functional sequence responsive to a corresponding boundary condition state of the concurrent shared object (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

56. MacGregor has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of MacGregor to maintain data coherency.

57. In addition, MacGregor has not taught at least one functional sequence implementing an access operation on a concurrent shared object, the concurrent shared object instantiable as a circular buffer of bounded size implementing a contiguous array delimited by a pair of end identifying indices. Weiss has taught at least one functional sequence implementing an access operation on a concurrent shared object, the concurrent shared object instantiable as a circular buffer of bounded size implementing a contiguous array delimited by a pair of end identifying indices (Weiss page 111). A person of ordinary skill in the art would have recognized, and as taught by Weiss, a circular queue lets elements be added to the beginning of a queue after they have been enqueued, thereby allowing more elements to enter the queue even after all positions have been used at one point. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Weiss in MacGregor.

58. Referring to claim 42, MacGregor has taught wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage

medium and a network, wireline, wireless or other communications medium (MacGregor column 3, lines 15-28 and Figure 1).

59. Claims 33-34 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacGregor in view of Greenspan and in view of Weiss as applied to claim 32 above, and further in view of Arnold, U.S. Patent Number 5,081,572 (herein referred to as Arnold). Arnold has taught

a. Wherein the first access operation includes a push (Applicant's claim 33) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

b. Wherein the first access operation includes a pop (Applicant's claim 34) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B); and

c. Wherein the at least one functional sequence includes opposing end variants of push and pop operations on the concurrent shared object (Applicant's claim 41) (MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B).

60. MacGregor has not taught

a. Wherein, on failure, the DCAS returns an indication by which an full state of the contiguous array is detected (Applicant's claim 33);

b. Wherein, on failure, the DCAS returns an indication by which an empty state of the contiguous array is detected (Applicant's claim 34);

c. Wherein the boundary condition state corresponding to push operations is a full state of the array (Applicant's claim 41); and

d. Wherein the boundary condition state corresponding to pop operations is an empty state of the array (Applicant's claim 41).

61. Arnold has taught

a. Wherein, on failure, the DCAS returns an indication by which a full state of the contiguous array is detected (Applicant's claim 33) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8);

b. Wherein, on failure, the DCAS returns an indication by which an empty state of the contiguous array is detected (Applicant's claim 34) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8);

c. Wherein the boundary condition state corresponding to push operations is a full state of the array (Applicant's claim 41) (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8); and

d. Wherein the boundary condition state corresponding to pop operations is an empty state of the array (Applicant's claim 41) (Arnold column 8, lines 28-34;

column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8).

62. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Arnold, identifying the empty/full state ensures the queue will not be corrupted, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the empty/full state of Arnold in MacGregor to ensure data coherency.

63. Claim 43 is rejected under 35 U.S.C. 103(a) as being unpatentable over Arnold, U.S. Patent Number 5,081,572 (herein referred to as Arnold) in view of Greenspan et al., U.S. Patent Number 6,128,710 (herein referred to as Greenspan) and in view of Mark Allen Weiss's Data Structures and Algorithm Analysis in C++ Second Edition © 1999 (herein referred to as Weiss). Arnold has taught an apparatus comprising:

a. Plural processors (Arnold column 1, lines 7-14);

b. A store addressable by each of the plural processors (Arnold column 1, lines 7-14);

c. First- and second-end index stores accessible to each of the plural processors for identifying opposing ends of a bounded-size contiguous array encoded in buffer form in the addressable store (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8); and

d. Means for coordinating competing access operations, the coordinating means employing in each instance thereof, at least one dual target compare and swap

(DCAS) operation to disambiguate a retry state and a boundary condition state of the array based on then-current contents of one, but not both, of first- and second end index stores and an array element corresponding thereto (Arnold column 8, lines 28-34; column 9, lines 23-34; column 11, lines 7-24; column 13, lines 7-13; Figure 4A; Figure 4B; Figure 6; and Figure 8).

64. Arnold has not taught an atomic compare and swap. Greenspan has taught atomic compare and swap (Greenspan column 3, line 42 to column 4, line 18). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Greenspan, locking instructions to prevent them from being interrupted maintains data coherency, thereby preventing data errors. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the atomic compare and swaps of Greenspan in the device of Arnold to maintain data coherency.

65. In addition, MacGregor has not taught a circular buffer form. Weiss has taught a circular buffer form (Weiss page 111). A person of ordinary skill in the art would have recognized, and as taught by Weiss, a circular queue lets elements be added to the beginning of a queue after they have been enqueued, thereby allowing more elements to enter the queue even after all positions have been used at one point. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Weiss in Arnold.

Response to Arguments

66. Applicant's arguments filed 09 August 2004 and 26 August 2004 have been fully considered but they are not persuasive.

67. Applicant argues in essence on pages 13-14

> "At least one indication of the art of record does not address the challenges of accessing a shared data structure...Arnold's technique allows a second thread of control to intervene after checking for an empty state, but before Arnold's compare and load is performed."

68. This has not been found persuasive. The claims have only recited that the dual target compare and swap is atomic, i.e. only the compare and swap cannot be interrupted. The claim language does not state that the combined operations of checking for an empty state and compare and swaps cannot be interrupted. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., not allowing a second thread to intervene after checking for an empty state) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

69. Applicant argues in essence on pages 14 and 15

> "...executing an atomic multi-target compare and swap operation to both access a data structure susceptible to concurrent access and to detect boundary condition states upon failure of the atomic multi-target compare and swap operation. The boundary condition state detection and the accessing of the data structure are constituent aspects of the atomic multi-target compare and swap, thus addressing challenges presented in environments with multiple threads of control (page 14)"

"Integrating access and boundary condition state detection for managing concurrent access to a data structure...

...emphasize utilization of Applicant's invention in a parallel system (page 15)..."

70. This has not been found persuasive. The concurrent access, when actually stated in the claim, is not claimed in the body of the claim, but in the preamble of the claim. The body of the claims contain detecting the boundary conditions and the atomic multi-target compare and swap operation, but not its relation to concurrent access to a data structure or multiple threads of control. Since there is nothing in the body relating back to the aspect of concurrent access in the preamble of the claim, there was not patentable weight give to the concurrent access. In response to applicant's arguments, the recitation concurrent access to a data structure has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

71. Also, for some claims, the concurrent access and parallel system is not explicitly claimed. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., concurrent access) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification,

limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

72. Applicant argues on page 15 "...the Office Action never identifies art that discloses or suggests 'a contiguous array S...a left index L and a right index R into the contiguous array...wherein at least the S[L] and S[R] entries encode a distinguishing value'..." This has not been found persuasive. Examiner has cited MacGregor Abstract; column 1, line 11 to column 2, line 14; column 2, lines 19-42; column 6, line 53 to column 7, line 24; column 8, line 39 to column 9, line 2; column 10, lines 19-64; Figure 3A; Figure 3B; Figure 4A; and Figure 4B as seen in the previous and above rejections. Should there be any questions relating to this, please contact the Examiner.

73. Applicant argues on page 15 in essence "The art of record discloses operations for singly linked lists and doubly linked lists, but does not evince techniques for accessing a concurrent shared object." This has not been found persuasive. Weiss has taught an accessible circular buffer, as shown on pages 110-115 provided with a previous Office Action. This circular buffer functions with the add and delete operations similar to the ones taught by Weiss on the aforementioned pages. These add and delete operations access the object.

74. Applicant argues on page 16 in essence "...does not address 'coordinating competing access operations.'" This has not been found persuasive. Arnold is for parallel processor systems and memory access instructions from these parallel systems (Arnold column 4, lines 16-28). This means competing access operations because these parallel systems access the same queues and Arnold ensures that no matter how many parallel systems access the queues, the queue integrity will hold, meaning the queue data will no become corrupt with bad data.

Conclusion

75. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

76. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

77. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (703) 305-7596. The examiner can normally be reached on M-T 7:30am-5:00pm.

78. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

79. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
April 19, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100